Curriculum

of

Bachelor of Science in Software Engineering

BS (SE)

Revised: Fall 2018

Implemented: Spring 2019

Department of Software Engineering,

School of Systems and Technology,

University of Management and Technology,

Lahore, Pakistan

1. Table of Contents

1. Curriculum Review Committee

Following are the members of the curriculum review committee who were involved in the revision of the BS-SE program.

1. Dr. Shaukat Iqbal
   Professor and Dean, School of Systems and Technology
2. Dr. Malik Tahir Hassan
   Assistant Professor and Chairperson, Department of Software Engineering
3. Dr. Tayyaba Anees
   Assistant Professor and Director, Department of Software Engineering
4. Mr. Muhammad Ahsan
   Assistant Professor and Program Advisor, Department of Software Engineering
5. Dr. Tahir Ejaz
   Assistant Professor and Director Final Year Projects, Department of Software Engineering
6. Mr. Mustansar Ali Khan
   Lecturer, Department of Software Engineering

2. Preface

The curriculum of Software Engineering is very dynamic and requires continuous revision. We have been revising our elective courses and their contents to cope with these dynamic requirements of the program. Major overhauling of the curriculum was done based on the feedback from academic and industry experts, our faculty members, and the students last year. This time, minor changes including implementation of separate labs for lab courses and update of reference content material have been done. The revised curriculum conforms to the guidelines by National Computing Education Accreditation Council (NCEAC) and Higher Education Commission (HEC) of Pakistan. We believe that the current BS (SE) curriculum will prove to be more recent, relevant, applied and standardized.

3. Acknowledgment

We are thankful to Allah Almighty for all his blessings, and enabling us complete this important task. We acknowledge and admire Dr. Muhammad Aslam, the honorable rector of UMT, for his vision, motivation and support, moral as well as logistic, to make our academic programs up-to-date by including state-of-the-art courses, materials and practices. We also acknowledge the support of our faculty members and members of BoS and BoF for sharing their invaluable experiences and pieces of advice to improve the curriculum.

4. Introduction

BS(SE) Program aims to create, expand, disseminate and teach the Software Engineering body of knowledge through academics, applications and research which positively impact society locally, nationally, and internationally.
The aim of the program is to develop a broad understanding of the discipline of software engineering. It seeks to complement this with a detailed knowledge of techniques for the analysis and design of complex software intensive systems. The basic objective of software engineering is to develop methods and procedures for software development that can scale up for large systems and that can be used consistently to produce high-quality software at low cost and with a small cycle of time.

a) Vision

The vision for our SE program is on imparting the knowledge and training which will enable students to harmonize theory with practice, concept with application, and problem with solution. And to prepare them to apply engineering principles, practices, and processes to design, develop, deploy, and maintain software systems. The program vision is the development of student's professional and interpersonal skills. The program will help students to analyze, manage and design complex software systems. It will help students to enhance their adaptability to team environments. The program will also strive to develop a capacity for innovation and a passion for lifelong learning.

b) Mission

Software engineering is the discipline of creating high-quality software systems in a systematic, controlled and efficient manner. It involves the application of engineering concepts, techniques, and methods to the design, development, deployment and maintenance of software systems. Mission for our Software engineering program is to develop professionals who have a mastery of principles, theory, practices, and processes necessary to produce quality software systems.

c) Core Values / Philosophy

In the early years of computing, there was no clear notion of software development methodology. Software was developed in an ad-hoc fashion. Initially, coding in machine language and eventually in high-level programming languages. The main stimulus for Software engineering was adopting some sort of methodological approach about how software should be developed due to the rapid complexity associated with Software Development.

2.

3. Objectives

After completing the program, a student will be able to:

- apply the basic principles of software engineering for providing reliable software.
- design, implement, deploy and maintain practical software systems.
- verify and validate software systems.
- understand and apply software project management skills: measurement, estimation, costing, planning, deployment and tracking of resources.
- communicate effectively in career and educational environments, and

- apply his/her knowledge and skills to develop a career in software business or industry, or for graduate study in software engineering or other scientific or technical fields, in general.

5. Program Structure

d) Category-wise course distribution

5.1 Computing Core (11 Courses, 39 credit hours)

1. Programming Fundamentals
2. Object Oriented Programming
3. Data Structures & Algorithms
4. Discrete Structures
5. Operating Systems
6. Database Systems
7. Software Engineering
8. Computer Networks
9. Information Security
10. Final Year Project I
11. Final Year Project II

5.2 Program Core (8 courses, 24 credit hours)

1. Human Computer Interaction
2. Software Construction & Development
3. Software Design & Architecture
4. Software Project Management
5. Software Quality Engineering
6. Software Re-Engineering
7. Software Requirements Engineering
8. Web Engineering

5.3 Mathematics and Science Foundation Courses (4 courses, 12 credit hours)

4. Applied Physics 3-0
5. Calculus & Analytical Geometry 3-0
6. Linear Algebra 3-0
7. Probability & Statistics

5.4 University Core (7 courses, 21 credit hours)

1. English Composition & Comprehension
2. Technical & Business Writing
3. Communication & Presentation Skills
4. Professional Practices
5. Intro to Info. & Comm. Technologies
6. Pakistan Studies
7. Islamic Studies/ Ethics

5.5 Program Electives (Technical Electives) Annexure I

5 Courses,15 credit hours

5.6 Supporting Electives Annexure II

3 Courses, 9 credit hours

5.7 University Electives Annexure III

4 Courses, 12 credit hours

Note: Elective courses can be offered from the list as required and decided every semester. The list is available on SST Website and is not exhaustive. Sample elective courses are presented in the Course Outlines section.

e) Comparison of our structure with HEC

Credit Hours Distribution (Our)

Major Areas	Core/Required	Elective	Credit Hours
Computing Foundation	39	15	78
Software Engineering	24		
Supporting Studies	12	9	21
General Education	21	12	33
Total			**132**

Credit Hours Distribution (HEC)

Major Areas	Core/Required	Elective	Credit Hours
Computing Foundation	39	15	78
Software Engineering	24		
Supporting Studies	12	9	21
General Education	19	12	31
Total			**130**

6. Semester-wise road map (Implemented from Sp2019, Batch 15)

Total courses: 42 Total Credit Hours: 132

1st Semester				2nd Semester			
Course Code	**Course Title**	**Cr. Hrs.**	**Pre-req**	**Course Code**	**Course Title**	**Cr. Hrs.**	**Pre-req**
IT1091	Introduction to Info & Comm. Technologies	2		CC1022	Object Oriented Programming	3	CC1021, CC1021L
IT1091L	Introduction to Info & Comm. Technologies Lab	1		CC1022L	Object Oriented Programming Lab	1	CC1021, CC1021L
NS125	Applied Physics	2		CC1041	Discrete Structures	3	
NS125L	Applied Physics Lab	1		EN125	Composition & Communication	3	EN111
MA100	Calculus and Analytical Geometry	3			Supporting Elective-I (e.g., SE1105-Business Process Engineering)	3	
EN111	English Grammar & Comprehension	3			GE/University Elective-I	3	
CC1021	Programming Fundamentals	3					
CC1021L	Programming Fundamentals Lab	1					
	Semester Credit Hours	**16**			**Semester Credit Hours**	**16**	

3rd Semester				4th Semester			
Course Code	**Course Title**	**Cr. Hrs.**	**Pre-req**	**Course Code**	**Course Title**	**Cr. Hrs.**	**Pre-req**
CC2042	Data Structures and Algorithms	3	CC1022, CC1022L	CC2141	Database Systems	3	CC2042, CC2042L
CC2042L	Data Structures and Algorithms Lab	1	CC1022, CC1022L	CC2141L	Database Systems Lab	1	CC2042, CC2042L
MA210	Linear Algebra	3			Supporting Elective-II(e.g., IT4052-OperationResearch)	3	
CC2101	Software Engineering	3		MA150	Probability and Statistics	3	
	GE/University Elective II	3		SE2102	Software Requirement Engineering (SRE)	3	CC2101
	GE/University Elective III	3			GE/University Elective IV	3	
POL101	Pakistan Studies	3					
	Semester Credit Hours	**19**			**Semester Credit Hours**	**16**	

5th Semester				6th Semester			
Course Code	**Course Title**	**Cr. Hrs.**	**Pre-req**	**Course Code**	**Course Title**	**Cr. Hrs.**	**Pre-req**
SE3103	Software Design and Architecture	3	SE2102	SE3111	Software Construction and Development	2	SE3103
CC3011	Operating Systems	3	SE2042, SE2042L	SE3111L	Software Construction and Development Lab	1	SE3103
CC3011L	Operating Systems Lab	1	SE2042, SE2042L	SE3162	Web Engineering	3	
	SE Elective I	3		CC3071	Computer Networks	3	
	SE Elective II	3		CC3071L	Computer Networks Lab	1	
	Supporting Elective III (e.g., SE3104-Formal Methods in Software Engineering)	3		CC3121	Information Security	3	
ISL101	Islamic Studies	3			SE Elective III	3	
	Semester Credit Hours	**19**			**Semester Credit Hours**	**16**	

7th Semester				8th Semester			
Course Code	**Course Title**	**Cr. Hrs.**	**Pre-req**	**Course Code**	**Course Title**	**Cr. Hrs.**	**Pre-req**
CC4181	Final Year Project-I	3		CC4182	Final Year Project-II	3	CC4181
SE4192	Software Project Management	3	CC2101	SE4114	Human Computer Interaction	3	CC2101
SE4112	Software Quality Engineering	3	CC2101	HU4092	Professional Practices	3	
	SE Elective IV	3			SE Elective V	3	
SE4113	Software Re-Engineering	3	SE3111, SE3111L				
EN220	Research Paper Writing & Presentation	3	EN125				
	Semester Credit Hours	**18**			**Semester Credit Hours**	**12**	

7. Course outlines

8.

Introduction to Information and Communication Technologies			
Credit Hours:	3 (2,1)	**Prerequisites:**	none

Course Learning Outcomes (CLOs):

The students will be able to recognize different parts of computer and will be given exposure to the application of computing in the real modern world. Also, they will be introduced to the concepts of problem-solving using computing technologies.

Course Content:

This course introduces the fundamentals of computers and information technology to beginners. Initially, the students are familiarized with the computer, its components and working. They are trained to use the computer and different common application software's. They are then taught the techniques of logic development for solving problems using pseudo-code. They are also introduced to the concepts of relational databases, computer networking and internetworking.

Teaching Methodology:

Lecturing, Written Assignments, Project, Report Writing, Final Exam

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Presentation, Final Exam

Reference Materials:

1. Computing Fundamentals, Peter Norton, McGrawHill, 6th Ed 2005
2. Introduction to Computers (Seventh Edition) Peter Norton. 2010.
3. Computing Fundamentals, Faithe Wempen, Sybex, 2015
4. Discovering Computers by Shelly 2016

English Composition & Comprehension			
Credit Hours:	3 (3,0)	**Prerequisites:**	none

Course Learning Outcomes (CLOs):

1. The students will be able to read and write University level English.
2. Write clearly and effectively in a variety of forms, adapting writing and analytical skills to all situations.
3. Adapt literary, critical and oral skills to communicate effectively in business and graduate school environments.

Course Content:

Paragraph and Essay Writing, Descriptive Essays; Sentence Errors, Persuasive Writing; How to give presentations, Sentence Errors; Oral Presentations, Comparison and Contrast Essays, Dialogue Writing, Short Story Writing, Review Writing, Narrative Essays, Letter Writing

Teaching Methodology:

Lecturing, Written Assignments, Presentation, Report Writing

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Presentation, Final Exam

Reference Materials:

1. College Writing Skills with Readings, by John Langan, McGraw-Hill, 9th Edition. 2013
2. A Textbook of English Prose and Structure by Arif Khattak, et al, GIKI Institute, 2000

Calculus & Analytical Geometry			
Credit Hours:	3 (3,0)	**Prerequisites:**	none

Course Learning Outcomes (CLOs):

The students will be able to use the basic concepts ***of analytic geometry*** and the elementary operations of ***calculus***.

Course Content:

Limits and Continuity; Introduction to functions, Introduction to limits, Techniques of funding limits, Indeterminate forms of limits, Continuous and discontinuous functions and their applications, Differential calculus; Concept and idea of differentiation, Geometrical and Physical meaning of derivatives, Rules of differentiation, Techniques of differentiation, Rates of change, Tangents and Normals lines, Chain rule, implicit differentiation, linear approximation, Applications of differentiation; Extreme value functions, Mean value theorems, Maxima and Minima of a function for single-variable, Concavity, Integral calculus; Concept and idea of Integration, Indefinite Integrals, Techniques of integration, Riemann sums and Definite Integrals, Applications of definite integrals, Improper integral, Applications of Integration; Area under the curve, Analytical Geometry; Straight lines in R3, Equations for planes.

Teaching Methodology:

Lecturing, Written Assignments

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Final Exam

Reference Materials:

1. Calculus and Analytic Geometry by Kenneth W. Thomas. 1992
2. Calculus by Earl William Swokowski; Michael Olinick; Dennis Pence; Jeffery A. Cole. 1994
3. Calculus by Stewart, James. 2011

Programming Fundamentals			
Credit Hours:	4 (3,1)	**Prerequisites:**	none

Course Learning Outcomes (CLOs):

1. Understand basic problem solving steps and logic constructs
2**. Apply** basic programming concepts
3. **Design** and implement algorithms to **solve** real world problems.

Course Content:

Introduction to problem solving, a brief review of Von-Neumann architecture, Introduction to programming, role of compiler and linker, introduction to algorithms, basic data types and variables, input/output constructs, arithmetic, comparison and logical operators, conditional statements and execution flow for conditional statements, repetitive statements and execution flow for repetitive statements, lists and their memory organization, multi-dimensional lists, introduction to modular programming, function definition and calling, stack rolling and unrolling, string and string operations, pointers/references, static and dynamic memory allocation, File I/O operations

Teaching Methodology:

Lectures, Written Assignments, Practical labs, Semester Project, Presentations

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Project, Presentations, Final Exam

Reference Materials:

1. Starting out with Python, 4th Edition, Tony Gaddis. 2017.
2. Starting out with Programming Logic & Designs, 4th Edition, Tony Gaddis, 2015.
3. Object Oriented Programming in C++ by Robert Lafore 2001.
4. C How to Program, 8th Edition by Paul Deitel & Harvey Deitel 2015.
5. Problem Solving and Program Design in C++, 7th Edition by Jeri R. Hanly & Elliot B. Koffman 2000.
6. Practice of Computing Using Python, 3rd Edition by William Punch & Richard Enbody, 2016.
7. Introduction to Computation and Programming Using Python: With Application to Understanding Data, 2nd Edition by Guttag, John 2016.

Applied Physics			
Credit Hours:	3 (3,0)	**Prerequisites:**	none

Course Learning Outcomes (CLOs):

At the end of this course the students will be able to apply the concepts of physics in real world situations

Course Content:

Electric force and its applications and related problems , conservation of charge, charge quantization, Electric fields due to point charge and lines of force. Ring of charge, Disk of charge, A point charge in an electric field, Dipole in a n electric field, The flux of vector field, The flux of electric field, Gauss' Law, Application of Gauss' Law, Spherically symmetric charge distribution, A charge isolated conductor, Electric potential energy, Electric potentials, Calculating the potential from the field and related problem Potential due to point and continuous charge distribution, Potential due to dipole, equipotential surfaces, Calculating the field from the potential , Electric current, Current density, Resistance, Resistivity and conductivity, Ohm's law and its applications, The Hall effect, The magnetic force on a current, The Biot- Savart law, Line of B, Two parallel conductors, Amperes' s Law, Solenoid, Toroids, Faraday's experiments, Faraday's Law of Induction, Lenz's law, Motional emf, Induced electric field, Induced electric fields, The basic equation of electromagnetism, Induced Magnetic field, The displacement current, Reflection and Refraction of light waves, Total internal reflection, Two source interference, Double Slit interference, related problems, Interference from thin films, Diffraction and the wave theory, related problems, Single-Slit Diffraction, related problems, Polarization of electromagnetic waves, Polarizing sheets, related problems.

Teaching Methodology:

Lecturing, Written Assignments, Project, Experiments, Report Writing

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Experiments, Final Exam

Reference Materials:

1. 1 Narciso Garcia, Arthur Damask, Steven Schwarz., "Physics for Computer Science Students", Springer Verlag, 1998..
2. Fundamentals of Physics (Extended), 10th edition, Resnick and Walker, 2013.

Object Oriented Programming			
Credit Hours:	4 (3,1)	**Prerequisites:**	Programming Fundamentals

Course Learning Outcomes (CLOs):

1. Understand principles of object oriented paradigm.
2. Identify the objects & their relationships to build object oriented solution
3. Model a solution for a given problem using object oriented principles
4. Examine an object oriented solution.

Course Content:

Introduction to object oriented design, history and advantages of object oriented design, introduction to object oriented programming concepts, classes, objects, data encapsulation, constructors, destructors, access modifiers, const vs non-const functions, static data members & functions, function overloading, operator overloading, identification of classes and their relationships, composition, aggregation, inheritance, multiple inheritance, polymorphism, abstract classes and interfaces, generic programming concepts, function & class templates, standard template library, object streams, data and object serialization using object streams, exception handling.

Teaching Methodology:

Lectures, Written Assignments, Practical labs, Semester Project, Presentations

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Project, Presentations, Final Exam

Reference Materials:

1. C++ How to Program, 10th Edition, Deitel & Deitel. 2016.
2. Object Oriented Programming in C++, 3rd Edition by Robert Lafore 1998.
3. Java: How to Program, 9th Edition by Paul Deitel, 2011.
4. Beginning Java 2, 7th Edition by Ivor Horton, 2011.
5. An Introduction to Object Oriented Programming with Java, 5th Edition by C. Thomas Wu, 2009.
6. Starting Out with C++ from Control Structures to Objects, 9th Edition, Tony Gaddis, 2017.

Discrete Structures			
Credit Hours:	3 (3,0)	**Prerequisites:**	none

Course Learning Outcomes (CLOs):

1. **Understand** the key concepts of Discrete Structures such as Sets, Permutations, Relations,
Graphs, and Trees etc.
2. **Apply** formal logic proofs and/or informal, but rigorous, logical reasoning to real problems, such as predicting the behavior of software or solving problems such as puzzles.
3. **Apply** discrete structures into other computing problems such as formal specification, verification, databases, artificial intelligence, and cryptography.
4. **Differentiate** various discrete structures and their relevance within the context of computer science, in the areas of data structures and algorithms, in particular.

Course Content:

Introduction to object oriented design, history and advantages of object oriented design, introduction to object oriented programming concepts, classes, objects, data encapsulation, constructors, destructors, access modifiers, const vs non-const functions, static data members & functions, function overloading, operator overloading, identification of classes and their relationships, composition, aggregation, inheritance, multiple inheritance, polymorphism, abstract classes and interfaces, generic programming concepts, function & class templates, standard template library, object streams, data and object serialization using object streams, exception handling.

Teaching Methodology:

Lectures, Written Assignments, Practical labs, Semester Project, Presentations

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Project, Presentations, Final Exam

Reference Materials:

1. Discrete Mathematics with Applications, 4th Edition by Susanna S. Epp, 2010.
2. Discrete Mathematics, 7th edition by Richard Johnson Baugh, 2007.
3. Discrete Mathematical Structures, 4th edition by Kolman, Busby & Ross, 1999.
4. Discrete and Combinatorial Mathematics: An Applied Introduction by Ralph P. Grimaldi, 2003.
5. Logic and Discrete Mathematics: A Computer Science Perspective by Winifred Grassman, 1995.
6. Discrete Mathematics and Its Applications, 7th edition by Kenneth H. Rosen, 2011.

Communication & Presentation Skills			
Credit Hours:	3 (3,0)	**Prerequisites:**	English Composition & Comprehension

Course Learning Outcomes (CLOs):

The students will get better exposure on how to interact with not only individuals but with organizations at different levels and how to present them professionally.

Course Content:

Principles of writing good English, understanding the composition process: writing clearly; words, sentence and paragraphs; Comprehension and expression; Use of grammar and punctuation. Process of writing, observing, audience collecting, composing, drafting and revising, persuasive writing, reading skills, listening skills and comprehension, skills for taking notes in class, skills for exams; Business communications; planning messages, writing concise but with impact. Letter formats, mechanics of business, letter writing, letters, memo and applications, summaries, proposals, writing resumes, styles and formats, oral communications, verbal and non-verbal communication, conducting meetings, small group communication, taking minutes. Presentation skills; presentation strategies, defining the objective, scope and audience of the presentation, material gathering material organization strategies, time management, opening and concluding, use of audio-visual aids, delivery and presentation.

Teaching Methodology:

Lecturing, Written Assignments, Project, Presentation, Report Writing

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Presentation, Final Exam

Reference Materials:

1. Practical Business English, Collen Vawdrey, 1993, ISBN = 0256192740
2. Effective Communication Skills: The Foundations for Change, John Nielsen, 2008, ISBN = 1453506748

Data Structures and Algorithms			
Credit Hours:	4 (3,1)	**Prerequisites:**	OOP

Course Learning Outcomes (CLOs):

1**. Implement** various data structures and their algorithms, and **apply** them in implementing
simple applications.
2. **Analyze** simple algorithms and determine their complexities.
3. **Apply** the knowledge of data structures to other application domains.
4. **Design** new data structures and algorithms to solve problems.

Course Content:

Abstract data types, complexity analysis, Big Oh notation, Stacks (linked lists and array implementations), Recursion and analyzing recursive algorithms, divide and conquer algorithms, Sorting algorithms (selection, insertion, merge, quick, bubble, heap, shell, radix, bucket), queue, dequeuer, priority queues (linked and array implementations of queues), linked list & its various types, sorted linked list, searching an unsorted array, binary search for sorted arrays, hashing and indexing, open addressing and chaining, trees and tree traversals, binary search trees, heaps, M-way tress, balanced trees, graphs, breadth-first and depth-first traversal, topological order, shortest path, adjacency matrix and adjacency list implementations, memory management and garbage collection.

Teaching Methodology:

Lectures, Written Assignments, Practical labs, Semester Project, Presentations

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Project, Presentations, Final Exam

Reference Materials:

1. Data Structures and Algorithms in C++ by Adam Drozdek, 2012.
2. Starting Out with Java: From Control Structures through Data Structures, 4th Edition, Tony Gaddis, Pearson; 4th Edition, 2018
3. Data Structures and Algorithm Analysis in Java by Mark A. Weiss, 2011.
4. Data Structures and Abstractions with Java by Frank M. Carrano & Timothy M. Henry, 2014.
5. Data Structures and Algorithm Analysis in C++ by Mark Allen Weiss, 2005.
6. Java Software Structures: Designing and Using Data Structures by John Lewis and Joseph Chase, 2013.

Probability & Statistics			
Credit Hours:	3 (3,0)	**Prerequisites:**	none

Course Learning Outcomes (CLOs):

At the end of this course the students will be able to apply the concepts of Probability and Statistics in real world situations.

Course Content:

Introduction to Statistics and Data Analysis, Statistical Inference, Samples, Populations, and the Role of Probability. Sampling Procedures. Discrete and Continuous Data. Statistical Modeling. Types of Statistical Studies. Probability: Sample Space, Events, Counting Sample Points, Probability of an Event, Additive Rules, Conditional Probability, Independence, and the Product Rule, Bayes' Rule. Random Variables and Probability Distributions. Mathematical Expectation: Mean of a Random Variable, Variance and Covariance of Random Variables, Means and Variances of Linear Combinations of Random Variables, Chebyshev's Theorem. Discrete Probability Distributions. Continuous Probability Distributions. Fundamental Sampling Distributions and Data Descriptions: Random Sampling, Sampling Distributions, Sampling Distribution of Means and the Central Limit Theorem. Sampling Distribution of S2, t-Distribution, F-Quantile and Probability Plots. Single Sample & One- and Two-Sample Estimation Problems. Single Sample & One- and Two-Sample Tests of Hypotheses. The Use of P-Values for Decision Making in Testing Hypotheses (Single Sample & One- and Two-Sample Tests), Linear Regression and Correlation. Least Squares and the Fitted Model, Multiple Linear Regression and Certain, Nonlinear Regression Models, Linear Regression Model Using Matrices, Properties of the Least Squares Estimators.

Teaching Methodology:

Lecturing, Written Assignments, Presentation, Final Exam

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Report Writing, Presentation, Final Exam

Reference Materials:

1. Probability and Statistics for Engineers and Scientists by Ronald E. Walpole, Raymond H. Myers, Sharon L. Myers and Keying E. Ye, Pearson; 9th Edition (January 6, 2011). ISBN-10: 0321629116

2. Probability and Statistics for Engineers and Scientists by Anthony J. Hayter, Duxbury Press; 3rd Edition (February 3, 2006), ISBN-10:0495107573

3. Schaum's Outline of Probability and Statistics, by John Schiller, R. Alu Srinivasan and Murray Spiegel, McGraw-Hill; 3rd Edition (2008). ISBN-10:0071544259

Professional Practices			
Credit Hours:	3 (3,0)	**Prerequisites:**	none

Course Learning Outcomes (CLOs):

Social Application of Ethics.

Course Content:

Computing Profession, Computing Ethics, Philosophy of Ethics. The Structure of Organizations, Finance and Accounting, Anatomy of a Software House, Computer Contracts, Intellectual Property Rights, The Framework of Employee Relations Law and Changing Management Practices, Human Resource Management and IT, Health and Safety at Work, Software Liability, Liability and Practice, Computer Misuse and the Criminal Law, Regulation and Control of Personal Information. Overview of the British Computer Society Code of Conduct, IEEE Code of Ethics, ACM Code of Ethics and Professional Conduct, ACM/IEEE Software Engineering Code of Ethics and Professional Practice. Accountability and Auditing, Social Application of Ethics.

Teaching Methodology:

Lecturing, Written Assignments, Presentation, Final Exam

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Report Writing, Presentation, Final Exam

Reference Materials:

1. Professional Issues in Software Engineering by Frank Bott, Allison Coleman, Jack Eaton and Diane Rowland, CRC Press; 3rd Edition (2000). ISBN-10: 0748409513
2. Computer Ethics by Deborah G. Johnson, Pearson; 4th Edition (January 3, 2009). ISBN-10: 0131112414
3. A Gift of Fire: Social, Legal, and Ethical Issues for Computing and the Internet (3rd Edition) by Sara Baase, Prentice Hall; 3rd Edition (2008). ISBN-10: 0136008488
4. Applied Professional Ethics by Gregory R. Beabout, University Press of America (1993). ISBN-10: 0819193747.

Software Engineering			
Credit Hours:	3 (3,0)	**Prerequisites:**	none

Course Learning Outcomes (CLOs):

1. Describe various software engineering processes and activities
2. Apply the system modeling techniques to model a medium size software system
3. Apply software quality assurance and testing principles to medium size software system.
4. Discuss key principles and common methods for software project management such as scheduling, size estimation, cost estimation and risk analysis.

Course Content:

Nature of Software, Overview of Software Engineering, Professional software development, Software engineering practice, Software process structure, Software process models, Agile software Development, Agile process models, Agile development techniques, Requirements engineering process, Functional and non-functional requirements, Context models, Interaction models, Structural models, behavioral models, model driven engineering, Architectural design, Design and implementation, UML diagrams, Design patterns, Software testing and quality assurance, Software evolution, Project management and project planning, configuration management, Software Process improvement.

Teaching Methodology:

Lecturing, Written Assignments, Project, Report Writing

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Project, Presentations, Final Exam

Reference Materials:

1. Software Engineering, Sommerville I., 10th Edition, Pearson Inc., 2014
2. Software Engineering, A Practitioner's Approach, Pressman R. S.& Maxim B. R., 8th Edition, McGraw-Hill, 2015.

Linear Algebra			
Credit Hours:	3 (3,0)	**Prerequisites:**	none

Course Learning Outcomes (CLOs):

1. How to analyze and solve a linear system of equations;
2. Important characteristics of matrices, such as its four fundamental subspaces, rank, determinant, eigenvalues and eigenvectors, different factorizations, etc;
3. How to use characteristics of a matrix to solve a linear system of equations or study properties of a linear transformation;
4. Important concepts of vector spaces such as independence, basis, dimensions, orthogonality, etc;
5. Properties of special categories of matrices such as symmetric, positive definite, etc;
6. Some applications of linear algebra in other branches of sciences, engineering, and economics.

Course Content:

Algebra of linear transformations and matrices. determinants, rank, systems of equations, vector spaces, orthogonal transformations, linear dependence, linear Independence and bases, eigenvalues and eigenvectors ,characteristic equations, Inner product space and quadratic forms

Teaching Methodology:

Lecturing, Written Assignments

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Final Exam

Reference Materials:

1. Elementary Linear Algebra by Howard Anton, 2010.
2. Linear Algebra and its Applications by Gibert Strang, 2006.

Software Requirements Engineering			
Credit Hours:	3 (3,0)	**Prerequisites:**	Software Engineering

Course Learning Outcomes (CLOs):

1. Describe the requirements engineering process
2. Effectively analyze software requirements for the development of cost-effective and efficient technical solutions.
3. Prepare both functional and non-functional requirements along with validation for a medium-size software system.
4. Document effective requirements in Software Requirements Specification (SRS) using clear, unambiguous requirements.

Course Content:

Introduction to Requirements Engineering, Software Requirements, classification of requirements, Requirements process, Levels/layers of requirements, Requirement characteristics, Analyzing quality requirements, Software requirements in the context of systems engineering, Requirement evolution, requirement traceability, requirement prioritization, trade-off analysis, risk analysis and impact analysis, Requirement management, interaction between requirement and architecture, Requirement elicitation, elicitation sources and techniques, Requirement specification and documentation, specification sources and techniques, Requirements validation and techniques, Management of Requirements, Introduction to Management, Requirements Management Problems , Managing Requirements in an Acquisition Organization, Supplier Organizations, Product Organizations, Requirements engineering for agile methods.

Teaching Methodology:

Lecturing, Written and Lab Assignments, Project, Report Writing

Course Assessment:

Sessional Exam, Home and Lab Assignments, Quizzes, Project, Presentations, Final Exam

Reference Materials:

1. Software Requirements, Wiegers K. &Beatty J., 3rd Ed. Microsoft Press, 2013
2. Requirements Engineering, Elizabeth Hull, Ken Jackson and Jeremy Dick. 3rd Ed, Springer-Verlag London Limited, 2011.
3. Requirements Engineering and Management for Software Development Projects, Chemuturi M., Springer New York, 2013.

Information Security			
Credit Hours:	3 (3,0)	**Prerequisites:**	none

Course Learning Outcomes (CLOs):

1. **Explain** key concepts of information security such as design principles, cryptography, risk management, and ethics
2. **Discuss** legal, ethical, and professional issues in information security.
3. **Apply** various security and risk management tools for achieving information security and privacy.
4. **Identify** appropriate techniques to tackle and solve problems in the discipline of information security.

Course Content:

Information security foundations, security design principles; security mechanisms, symmetric and asymmetric cryptography, encryption, hash functions, digital signatures, key management, authentication and access control; software security, vulnerabilities and protections, malware, database security; network security, firewalls, intrusion detection; security policies, policy formation and enforcement, risk assessment, cybercrime, law and ethics in information security, privacy and anonymity of data.

Teaching Methodology:

Lectures, Written Assignments, Semester Project, Presentations

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Project, Presentations, Final Exam

Reference Materials:

1. Computer Security: Principles and Practice, 4th edition by William Stallings, 2017.
2. Principles of Information Security, 6th edition by M. Whitman and H. Mattord, 2017.
3. Computer Security, 3rd edition by Dieter Gollmann, 2011.
4. Computer Security Fundamentals, 3rd edition by William Easttom, 2016.
5. Official (ISC)2 Guide to the CISSP CBK, 3rd edition, 2012.

Computer Networks			
Credit Hours:	4 (3,1)	**Prerequisites:**	none

Course Learning Outcomes (CLOs):

1. **Describe** the key terminologies and technologies of computer networks
2. **Explain** the services and functions provided by each layer in the Internet protocol stack.
3. **Identify** various internetworking devices and protocols, and their functions in a network.
4. **Analyze** working and performance of key technologies, algorithms and protocols.
5. **Build** Computer Network on various Topologies

Course Content:

Introduction and protocols architecture, basic concepts of networking, network topologies, layered architecture, physical layer functionality, data link layer functionality, multiple access techniques, circuit switching and packet switching, LAN technologies, wireless networks, MAC addressing, networking devices, network layer protocols, IPv4 and IPv6, IP addressing, sub netting, CIDR, routing protocols, transport layer protocols, ports and sockets, connection establishment, flow and congestion control, application layer protocols, latest trends in computer networks.

Teaching Methodology:

Lectures, Written Assignments, Practical labs, Semester Project, Presentations

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Project, Presentations, Final Exam

Reference Materials:

1. Computer Networking: A Top-Down Approach Featuring the Internet, 7th edition by James F. Kurose and Keith W. Ross, 2017.
2. Computer Networks, 5th Edition by Andrew S. Tanenbaum, 2010.
3. Data and Computer Communications, 10th Edition by William Stallings, 2013.
4. Data Communication and Computer Networks, 5th Edition by Behrouz A. Forouzan, 2012.

Database Systems			
Credit Hours:	4 (3,1)	**Prerequisites:**	DSA

Course Learning Outcomes (CLOs):

1. **Explain** fundamental database concepts.
2. **Design** conceptual, logical and physical database schemas using different data models.
3. **Identify** functional dependencies and resolve database anomalies by normalizing database tables.
4. **Use** Structured Query Language (SQL) for database definition and manipulation in any DBMS

Course Content:

Basic database concepts, Database approach vs file based system, database architecture, three level schema architecture, data independence, relational data model, attributes, schemas, tuples, domains, relation instances, keys of relations, integrity constraints, relational algebra, selection, projection, Cartesian product, types of joins, normalization, functional dependencies, normal forms, entity relationship model, entity sets, attributes, relationship, entity-relationship diagrams, Structured Query Language (SQL), Joins and sub-queries in SQL, Grouping and aggregation in SQL, concurrency control, database backup and recovery, indexes, NoSQL systems.

Teaching Methodology:

Lectures, Written Assignments, Practical labs, Semester Project, Presentations

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Project, Presentations, Final Exam

Reference Materials:

1. Database Systems: A Practical Approach to Design, Implementation, and Management, 6th Edition by Thomas Connolly and Carolyn Begg, 2014.
2. Database Systems: The Complete Book, 2nd Edition by Hector Garcia-Molina, Jeffrey D. Ullman, Jennifer Widom, 2008.
3. Database System Concepts, 6th Edition by Avi Silberschatz, Henry F. Korth and S. Sudarshan. 2010.
4. Database Management Systems, 3rd Edition by Raghu Ramakrishnan, Johannes Gehrke, 2002

Islamic Studies			
Credit Hours:	3 (3,0)	**Prerequisites:**	none
Course Learning Outcomes (CLOs):			
The students will be familiarized with the concepts of teachings of Islam via Quran and Hadith knowledge.			
Course Content:			
Basic Themes of Quran, Introduction to Sciences of Hadith, Introduction to Islamic Jurisprudence, Primary & Secondary Sources of Islamic Law, Makken & Madnian life of the Prophet, Islamic Economic System, Political theories, Social System of Islam			
Teaching Methodology:			
Lecturing, Written Assignments, Project			
Course Assessment:			
Sessional Exam, Home Assignments, Quizzes, Presentation, Final Exam			
Reference Materials: 1. Introduction to Islam by Dr Hamidullah, Papular Library Publishers Lahore, 1992. 2. Principles of Islamic Jurisprudence by Ahmad Hassan, Islamic Research Institute, IIUI, 1993. 3. Muslim Jurisprudence and the Quranic Law of Crimes, By Mir Waliullah, Islamic Books Services, 1992.			

Operating Systems			
Credit Hours:	4 (3,1)	**Prerequisites:**	DSA

Course Learning Outcomes (CLOs):

1. **Understand** the characteristics of different structures of the Operating Systems and identify the core functions of the Operating Systems.
2. **Analyze** and **evaluate** the algorithms of the core functions of the Operating Systems and explain the major performance issues with regard to the core functions.
3. **Demonstrate** the knowledge in applying system software and tools available in modern operating systems.

Course Content:

Operating systems basics, system calls, process concept and scheduling, inter-process communication, multithreaded programming, multithreading models, threading issues, process scheduling algorithms, thread scheduling, multiple-processor scheduling, synchronization, critical section, synchronization hardware, synchronization problems, deadlocks, detecting and recovering from deadlocks, memory management, swapping, contiguous memory allocation, segmentation & paging, virtual memory management, demand paging, thrashing, memory-mapped files, file systems, file concept, directory and disk structure, directory implementation, free space management, disk structure and scheduling, swap space management, system protection, virtual machines, operating system security

Teaching Methodology:

Lectures, Written Assignments, Practical labs, Semester Project, Presentations

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Project, Presentations, Final Exam

Reference Materials:

1. Operating Systems Concepts, 10th edition by Abraham Silberschatz, 2018.
2. Modern Operating Systems, 4th edition by Andrew S. Tanenbaum, 2014.
3. Operating Systems, Internals and Design Principles, 9th edition by William Stallings, 2017.

Technical & Business Writing			
Credit Hours:	3 (3,0)	**Prerequisites:**	Communication & Presentation Skills

Course Learning Outcomes (CLOs):

At the end of this course the students will be able to write clearly and effectively in a variety of forms, adapting writing and analytical skills to all situations.

Course Content:

Overview of technical reporting, use of library and information gathering, administering questionnaires, reviewing the gathered information; Technical exposition; topical arrangement, exemplification, definition, classification and division, casual analysis, effective exposition, technical narration, description and argumentation, persuasive strategy, Organizing information and generation solution: brainstorming, organizing material, construction of the formal outline, outlining conventions, electronic communication, generation solutions. Polishing style: paragraphs, listening sentence structure, clarity, length and order, pomposity, empty words, pompous vocabulary, document design: document structure, preamble, summaries, abstracts, table of contents, footnotes, glossaries, cross-referencing, plagiarism, citation and bibliography, glossaries, index, appendices, typesetting systems, creating the professional report; elements, mechanical elements and graphical elements. Reports: Proposals, progress reports, Leaflets, brochures, handbooks, magazines articles, research papers, feasibility reports, project reports, technical research reports, manuals and documentation, thesis. Electronic documents, Linear verses hierarchical structure documents.

Teaching Methodology:

Lecturing, Written Assignments, Presentation, Report Writing, Final Exam

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Report Writing, Presentation, Final Exam

Reference Materials:

1. Technical Report Writing, by Pauley and Riordan, Houghton Mifflin Company, 8th Ed. 2001
2. Effective Technical Communication by Ashraf Rizvi, Tata McGraw-Hill, 2005.

Operations Research			
Credit Hours:	3 (3,0)	**Prerequisites:**	none

Course Learning Outcomes (CLOs):

1. Learn the characteristics of different types of decision-making environments, appropriate decision making approaches and tools to be used in each type.
2. Solve the Transportation Models and Assignment Models.
3. Understand the basic methodology for the solution of linear programs and integer programs.

Course Content:

Introduction to operations research, History of operations research, Applications, Modeling the linear programming, Linear programming, Geometry, Solving the linear programming, the Simplex method, Shadow price, Theory of the simplex method, Duality, Dual theory, Sensitivity analysis, Other algorithms for linear programming, The dual simple method, Big – M method, The tow phase method, The transportation and assignment problems, The transportation problem, A streamlined simplex method for transportation problem, The assignment problem, A special algorithm for the assignment problem, Dynamic programming, Characteristic of dynamic programming, Deterministic dynamic programming, Integer programming, Prototype examples, BIP applications and formulation examples, Some perspectives on solving integer programming problems, The branch-and-cut approach to solve BIP problems, The incorporation of constraint programming.

Teaching Methodology:

Lectures, Written Assignments, Practical labs, Semester Project, Presentations

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Project, Presentations, Final Exam

Reference Materials:

1. Frederick S. Hiller, Gerald J. Lieberman, Introduction to Operations Research, 9th Edition, English, McGraw-Hill, 2010.
2. Operations Research: Applications and Algorithms, Wayne L Winston, Indian University, 4th edition, 2004

Professional Practices			
Credit Hours:	3 (3,0)	**Prerequisites:**	none

Course Learning Outcomes (CLOs):

1 Understand professional issues in Software Engineering
2 Understand Computer Ethics
3 Apply Professional Ethics

Course Content:

Computing Profession, Computing Ethics, Philosophy of Ethics. The Structure of Organizations, Finance and Accounting, Anatomy of a Software House, Computer Contracts, Intellectual Property Rights, The Framework of Employee Relations Law and Changing Management Practices, Human Resource Management and IT, Health and Safety at Work, Software Liability, Liability and Practice, Computer Misuse and the Criminal Law, Regulation and Control of Personal Information. Overview of the British Computer Society Code of Conduct, IEEE Code of Ethics, ACM Code of Ethics and Professional Conduct, ACM/IEEE Software Engineering Code of Ethics and Professional Practice. Accountability and Auditing, Social Application of Ethics.

Teaching Methodology:

Lecturing, Written Assignments, Presentation, Final Exam

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Report Writing, Presentation, Final Exam

Reference Materials:

1. Professional Issues in Software Engineering by Frank Bott, Allison Coleman, Jack Eaton and Diane Rowland, CRC Press; 3rd Edition (2000). ISBN-10: 0748409513
2. Computer Ethics by Deborah G. Johnson, Pearson; 4th Edition (January 3, 2009). ISBN-10: 0131112414
3. A Gift of Fire: Social, Legal, and Ethical Issues for Computing and the Internet (3rd Edition) by Sara Baase, Prentice Hall; 3rd Edition (2008). ISBN-10: 0136008488
4. Applied Professional Ethics by Gregory R. Beabout, University Press of America (1993). ISBN-10: 0819193747.

Pakistan Studies			
Credit Hours:	3 (3,0)	**Prerequisites:**	none

Course Learning Outcomes (CLOs):

Learn the history of Pakistan and study the current affairs of Pakistan

Course Content:

Historical background of Pakistan: Muslim society in Indo-Pakistan, the movement led by the societies, the downfall of Islamic society, the establishment of British Raj- Causes and consequences. Political evolution of Muslims in the twentieth century: Sir Syed Ahmed Khan; Muslim League; Nehru; Allama Iqbal: Independence Movement; Lahore Resolution; Pakistan culture and society, Constitutional and Administrative issues, Pakistan and its geo-political dimension, Pakistan and International Affairs, Pakistan and the challenges ahead.

Teaching Methodology:

Lectures, Written Assignments, Semester Project, Presentations

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Project, Presentations, Final Exam

Reference Materials:

1. The Emergence of Pakistan, Chaudary M., 1967
2. The making of Pakistan, Aziz. 1976
3. A Short History of Pakistan, I. H. Qureshi, ed., Karachi, 1988
4. Pakistan in the Twentieth Century, Lawrence Ziring ,ISBN: 9780195792768, 1999

Business Process Engineering			
Credit Hours:	3 (3,0)	**Prerequisites:**	none
Course Learning Outcomes (CLOs):			
Business process management, Manufacturing and services processes, Modelling and charting tools, Lean processes Improvement workshop techniques, Business process outsourcing, Re-engineering and improvement cases			
Course Content:			
Business process management, Manufacturing and services processes, Modelling and charting tools, Lean processes Improvement workshop techniques, Business process outsourcing, Re-engineering and improvement cases			
Teaching Methodology:			
Lecturing, Written Assignments, Project, Report Writing			
Course Assessment:			
Session Exam, Home Assignments, Quizzes, Final Exam			
Reference Materials: 1. Business Process Improvement; The Breakthrough Strategy for Total Quality, Productivity, and Competitiveness, H. J. Harrington, 1991 **2.** Business Intelligence: A Managerial Approach by Turban, Sharda, Delen, King, 2nd Edition, Prentice Hall ISBN: 13-978-0-136-10066-9 , 2011			

Business Process Management			
Credit Hours:	3 (3,0)	**Prerequisites:**	none

Course Learning Outcomes (CLOs):

1. Discover the various technologies that support Business Process Management
2. Analyze the performance of existing processes and identify process improvement.
3. Propose business solutions in written and verbal forms for process innovation and redesign projects.
4. Create a BPM implementation strategy and implementation plan for an organization.

Course Content:

Introduction to Business Process Management, Motivation and Definitions, Business Process Lifecycle, Classification of Business Processes, Goals, Structure, and Organization. Evolution of Enterprise Systems Architectures. Business Process Modeling. Process Orchestrations. Process Choreographies. Modeling in BPMN. Properties of Business Processes. Workflow Management Architectures, Flexible Workflow Management, Web Services and their Composition, Advanced Service Composition, Data-Driven Processes. Business Process Management Methodology.

Teaching Methodology:

Lecturing, Written Assignments, Project, Report Writing

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Final Exam

Reference Materials:

1. Business Process Management: Concepts, Languages, Architectures by Mathias Weske, Springer; 2nd Ed. 2012
2. Business Process Management Common Body of Knowledge by Yvonne LedererAntonucci, et. al., Create Space Independent Publishing Platform, 2009
3. Process Management: A Guide for the Design of Business Processes by Jörg Becker, Martin Kugeler and Michael Rosemann, Springer; 2nd Ed. 2011
4. BPMN Method and Style with BPMN Implementer's Guide: A structured approach for business process modeling and implementation using BPMN 2.0 by Bruce Silver, Cody Cassidy Press, 2011.

Computer Graphics			
Credit Hours:	3 (3,0)	**Prerequisites:**	none

Course Learning Outcomes (CLOs):

1. Comprehend the structure of modern computer graphics systems
2. Explain the basic principles of implementing computer graphics fundamentals
3. Compare key algorithms for modelling and rendering graphical data
4. Develop design and problem solving skills with applications to computer graphics
5. Construct interactive computer graphics programs using OpenGL

Course Content:

Fundamental Concepts: forward and backward rendering (i.e., ray-casting and rasterization), applications of computer graphics: including game engines, cad, visualization, virtual reality, polygonal representation, basic radiometry, similar triangles, and projection model, use of standard graphics APIs (see HCI GUI construction); basic rendering: rendering in nature, i.e., the emission and scattering of light and its relation to numerical integration, affine and coordinate system transformations, ray tracing, visibility and occlusion, including solutions to this problem such as depth buffering, painter's algorithm, and ray tracing, the forward and backward rendering equation, simple triangle rasterization, rendering with a shader-based API, texture mapping, including minification and magnification (e.g., trilinear MIP-mapping), application of spatial data structures to rendering, sampling and anti-aliasing, scene graphs and the graphics pipeline; geometric modeling: basic geometric operations such as intersection calculation, proximity tests, polynomial curves and surfaces, approximation techniques such as polynomial curves, bezier curves, spline curves and surfaces, animation as a sequence of still images.

Teaching Methodology:

Lecturing, Written Assignments, Project, Report Writing

Course Assessment:

Midterm exam, Final Exam, Assignments

Reference Materials:

1. Computer Graphics with Open GL (4th Edition) by Donald D. Hearn, Prentice Hall, 2010, ISBN-10: 0136053580.
2. Foundations of 3D Computer Graphics by S. J. Gortler, The MIT press, 2012.
3. Fundamentals of Computer Graphics, 3rd Edition, A K Peters, 2009.
4. Computer Graphics: Principles and Practice, 3rd Edition, Addison Wesley, 2013.
5. Real-Time Rendering, 3rd Edition, A K Peters, 2008.
Martin Kugeler and Michael Rosemann, Springer; 2nd Ed. 2011

Computer Vision			
Credit Hours:	3 (3,0)	**Prerequisites:**	none

Course Learning Outcomes (CLOs):

1. Understand and explain the field of computer vision in general for different applications, etc.
2. Understand and implement camera calibration
3. Work under OpenCV or Matlab computer vision toolbox, etc.
4. Implement an algorithm to assemble the extracted features to develop a higher-level perception
5. Implement different algorithms for spatial and frequency domain filtering, feature detection, structure from motion, motion estimation, etc.
6. To detect, recognize and track different types of the objects in the scene
7. Develop an algorithm for context awareness or scene understanding

Course Content:

Introduction, Image formation, Spatial and frequency domain processing, Feature detection and extraction, Image registration, Segmentation, Camera calibration, Structure from motion, Motion estimation, Stereo vision, Object detection and recognition, Object tracking, 3D scene reconstruction, Context and scene understanding, Image stitching, Image-based and video-based rendering, High-performance computing paradigms for vision and image processing.,

Teaching Methodology:

Lecturing, Written Assignments, Project, Report Writing

Course Assessment:

Midterm exam, Final Exam, Assignments

Reference Materials:

1. Computer Vision - A Modern Approach, by D. Forsyth and J. Ponce, Prentice Hall, 2003.
2. Szeliski R., Computer Vision - Algorithms and Applications, Springer, 2011.
3. J. R. Parker, Algorithms for Image Processing and Computer Vision, Willey Publishing Inc. 2011.
4. Gonzalez R. C., Woods R. E., Digital Image Processing, Pearson Education, 3rd edition, 2008.

Differential Equations			
Credit Hours:	3 (3,0)	**Prerequisites:**	none

Course Learning Outcomes (CLOs):

1. Identify, analyze and subsequently solve physical situations whose behavior can be described by ordinary differential equations.
2. Determine solutions to first order separable differential equations.
3. Determine solutions to first order linear differential equations.
4. Determine solutions to first order exact differential equations.
5. Determine solutions to second order linear homogeneous and non-homogeneous differential equations with constant coefficients.

Course Content:

Ordinary Differential Equations of the First Order: Geometrical Considerations, Isoclines, Separable Equations, Equations Reducible to Separable Form, Exact Differential Equations, Integrating Factors, Linear First-Order Differential Equations, variation of Parameters. Ordinary Linear Differential Equations; Homogeneous Linear Equations of the Second Order, Homogeneous Second-Order Equations with Constant Coefficients, General Solution, Real Roots, Complex Roots, Double Root of the Characteristic Equation, Differential Operators, Cauchy Equation, Homogeneous Linear Equations of Arbitrary Order, Homogeneous Linear Equations of Arbitrary Order with Constant Coefficients, Non- homogeneous Linear Equations. Modelling of Electrical Circuits. Systems of Differential Equations. Series Solutions of Differential Equations. Partial Differential Equations: Method of Separation of variables, wave, Heat & Laplace equations and their solutions by Fourier series method.

Teaching Methodology:

Lecturing, Written Assignments, Project, Report Writing

Course Assessment:

Midterm exam, Final Exam, Assignments

Reference Materials:

1. Advanced Engineering Mathematics Michael, G.1996, Prentice Hall Publishers.
2. Advanced Engineering Mathematics, 7th edition, Erwin, K. 1993, John Wiley & Sons Inc.
3. A First Course in Differential Equation Zill. Prindle. Weber. Schmidt.1996. Brooks/Cole Publishing.
4. Differential Equations with Boundary-Value Problems, Dennis. G. Zill, Michael, R. Cullen. 1996, Brooks/Cole Publishing,
5. Elementary Differential Equations with Applications C. H. Edwards. David, E. 1993. Penney, Prentice Hall.

DIGITAL IMAGE PROCESSING			
Credit Hours:	4(3,1)	**Prerequisites:**	none

Course Learning Outcomes (CLOs):

1. Understand the basics, applications in general, working inside the digital camera, sampling and quantization, image representation, etc.
2. Implement image enhancement, image segmentation, image transformations, spatial and frequency domain processing, filtering, convolution, image registration, feature detection, pattern recognition, etc.
3. Evaluate the performance of different image processing algorithms.

Course Content:

The human visual system, electromagnetic system, working and components inside digital camera, pixels, image representation, sampling, quantization, mathematics of image formation, convolution, camera projection, point-based image processing, Fourier theory, image filtering in spatial and frequency domain, wavelets, image registration, morphological operations, color models, multispectral images, feature detection, image segmentation, Pattern recognition, etc.

Teaching Methodology:

Lecturing, Written Assignments, Project, Report Writing

Course Assessment:

Midterm exam, Final Exam, Assignments

Reference Materials:

1. Gonzalez R. C., Woods R. E., Eddins S. L., Digital Image Processing Using Matlab, Pearson Education, 2nd edition, 2009.
2. Gonzalez R. C., Woods R. E., Digital Image Processing, Pearson Education, 3rd edition, 2008.
3. Understanding Digital Signal Processing by Richard G. Lyons, Prentice Hall; 3rd edition, 2010.

Formal Methods in Software Engineering			
Credit Hours:	3 (3,0)	**Prerequisites:**	none

Course Learning Outcomes (CLOs):

1. Describe the costs and benefits of formal methods'
2. Construct formal models of sequential software systems
3. Implement sequential software systems based on formal models
4. Verify attributes of formal models
5. Demonstrate formal correctness of simple procedure

Course Content:

Introduction to the use of mathematical models for specification and validation, Finite state machine models, models of concurrent systems, verification of models, and limitations. Analyzing well-formedness (e.g. completeness, consistency, robustness, etc.), Analyzing correctness (e.g. static analysis, simulation, model checking, etc.), Formal analysis, An introduction to VDM-SL, Sets, Sequences, Composite objects, Maps, VDM-SL, Comparative Formal Methods, Proofs, Introduction to Z

Teaching Methodology:

Lecturing, Written Assignments, Project, Report Writing

Course Assessment:

Midterm exam, Final Exam, Assignments

Reference Materials:

1. Modern Formal Methods and Applications, Hossam A. Gabbar, Springer-Verlag 2006.
2. Formal Software Development: From VDM to Java, Charatan, Quentin, and Aaron Kans. Palgrave Macmillan, 2003.
3. Understanding Z: a Specification Language and its Formal Semantics. J. M. Spivey. 1988. Cambridge University Press, New York, NY, USA.

Human Computer Interaction			
Credit Hours:	3 (3,0)	**Prerequisites:**	none

Course Learning Outcomes (CLOs):

At the end of the course the students will be able to:
1. Explain context of HCI and different measures for evaluation.
2. Apply the principles of good design for people from the perspective of age and disabilities.
3. Analyze techniques for user centered design for a medium sized software.
4. Evaluate the usability of a medium size software user interface.

Course Content:

Contexts for HCI, Psychology of usable things, Processes for User-Centered Design, Metrics and Measures for Evaluation, Usability heuristics and principles of Usability testing, Physical capabilities, Cognitive and social models for interaction design, Principles of good interaction design, Accessibility, Principles of GUI, Visual design elements, Data gathering, Task analysis, Prototyping, Help and user documentation, Internationalization, Usability inspection methods, Usability testing methods, New Interaction Technologies, Usability in practice, Visual Design and Typography, Icon Design, Ubiquitous, Augmented and Virtual Reality.

Teaching Methodology:

Lecturing, Written Assignments, Project, Report Writing

Course Assessment:

Midterm exam, Final Exam, Assignments

Reference Materials:

1. Designing the User Interface: Strategies for Effective Human-Computer Interaction, Ben Shneiderman and Catherine Plaisant, 6th Ed, Pearson Inc, 2016.
2. Designing Interactive Systems: A Comprehensive Guide to HCI, UX and Interaction Design, Benyon, D. 3rd Ed., Pearson. 2013
3. About Face: The Essentials of Interaction Design, Alan Cooper, Robert Reimann, David Cronin, Christopher Noessel, 4th Ed, Wiley, 2014

Mobile Application Development			
Credit Hours:	3 (3,0)	**Prerequisites:**	none

Course Learning Outcomes (CLOs):

1. Discuss different architectures & framework for Mobile Application development.
2. Develop mobile applications using current software development environments.
3. Compare the different performance tradeoffs in mobile application development.

Course Content:

Mobiles Application Development Platform; HTML5 for Mobiles; Android OS: Architecture, Framework and Application Development; iOS: Architecture, Framework; Application Development with Windows Mobile; Eclipse; Fragments; Calling Built-in Applications using Intents; Displaying Notifications; Components of a Screen; Adapting to Display Orientation; Managing Changes to Screen Orientation; Utilizing the Action Bar; Creating the User Interface; Listening for UI Notifications; Views; User Preferences; Persisting Data; Sharing Data; Sending SMS Messages; Getting Feedback; Sending E-mail; Displaying Maps; Consuming Web Services Using HTTP; Web Services: Accessing and Creating; Threading; Publishing, Android Applications; Deployment on App Stores; Mobile Programming Languages; Challenges with Mobility and Wireless Communication; Location-aware Applications; Performance/Power Tradeoffs; Mobile Platform Constraints; Emerging Technologies..

Teaching Methodology:

Lecturing, Written Assignments, Project, Report Writing

Course Assessment:

Midterm exam, Final Exam, Assignments

Reference Materials:

1. Professional Android application development, Reto Meier, Wrox Programmer to Programmer, 2015.
2. iOS Programming: The Big Nerd Ranch Guide, Conway, J., Hillegass, A., & Keur, C., 5th Edition, 2014.
3. Android Programming: The Big Nerd Ranch Guides, Phillips, B. & Hardy, B., 2nd Edition, 2014.

Natural Language Processing			
Credit Hours:	3 (3,0)	**Prerequisites:**	none

Course Learning Outcomes (CLOs):

1. Identify techniques for information retrieval, language translation, and text classification.
2. List the advantages of using standard corpora. Identify examples of current corpora for a variety of NLP tasks.
3. Define and contrast deterministic and stochastic grammars, providing examples to show the adequacy of each.
4. Simulate, apply, or implement classic and stochastic algorithms for parsing natural language.

Course Content:

Deterministic and stochastic grammars, Parsing algorithms, CFGs, Representing meaning / Semantics, Semantic roles, Temporal representations, Corpus-based methods, N-grams and HMMs, Smoothing and backoff, POS tagging and morphology, Information retrieval, Vector space model, Precision and recall, Information extraction, Language translation, Text classification, categorization, Bag of words model.

Teaching Methodology:

Lecturing, Written Assignments, Project, Report Writing

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Project, Presentations, Final Exam

Reference Materials:

1. Python Machine Learning, Sebastian Raschka. Publisher: Packt Publishing, 2015.
2. Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit Latest Edition, Steven Bird, Ewan Klein and Edward Loper Publisher: O'Reilly Media, 2009.
3. Speech and Language Processing, Latest Edition, Daniel Jurafsky and James H. Martin Publisher: Prentice Hall, 2000.

Software Construction and Development			
Credit Hours:	3 (2,1)	**Prerequisites:**	Software Design and Architecture

Course Learning Outcomes (CLOs):

1. Understand the role of design and its major activities within the OO software development process, with focus on the Unified process
2. Develop Object-oriented design models and refine them to reflect implementation details
3. Evaluate different architectures for a medium size software.
4. Implement design model using an object-oriented programming language.

Course Content:

Software development process, Software engineering process infrastructure, Software engineering process improvement, Systems engineering life cycle models, Process implementation, Levels of process definition, Life cycle model characteristics, Individual and team software process, Lehman's Laws, code salvaging, and configuration management. Martin Fowler's refactoring concepts and their application to small projects. Apply Michael Feathers' "legacy code" concepts. Exception handling, making methods robust by having them check their inputs sent from calling objects. Software configuration management, Release management, Software configuration management processes, Software deployment processes, Distribution and backup, Evolution processes and activities, Basic concepts of evolution and maintenance, Working with legacy systems, Refactoring, Error handling, exception handling, and fault tolerance. Personal reviews (design, code, etc.), Peer reviews (inspections, walkthroughs, etc.).

Teaching Methodology:

Lecturing, Written Assignments, Project, Report Writing

Course Assessment:

Midterm exam, Final Exam, Assignments

Reference Materials:

1. Clean Code: A Handbook of Agile Software Craftsmanship, Robert C. Martin, Prentice Hall, 2008.
2. The Pragmatic Programmer: From Journeyman to Master, Andrew Hunt and David Thomas, Addison-Wesley Professional, 1999.
3. Working Effectively with Legacy Code, Michael C. Feathers. Pearson Education, Prentice-Hall, 2004.
4. Refactoring: Improving the Design of Existing Code, Martin Fowler, Addison-Wesley Professional. 1999.

Web Engineering			
Credit Hours:	3 (3,0)	**Prerequisites:**	none

Course Learning Outcomes (CLOs):

CLO-1: Discuss how web standards impact software development.
CLO-2: Describe the constraints that the web puts on developers.
CLO-3: Design and Implement a simple web application.
CLO-4: Review an existing web application against a current web standard.

Course Content:

Web programming languages (e.g., HTML5, CSS 3, Java Script, PHP/JSP/ASP.Net), Design principles of Web based applications, Web platform constraints, Software as a Service (SaaS), Web standards, Responsive Web Design, Web Applications, Browser/Server Communication, Storage Tier, Cookies and Sessions, Input Validation, Full stack state management, Web App Security - Browser Isolation, Network Attacks, Session Attacks, Large scale applications, Performance of Web Applications, Data Centers, Web Testing and Web Maintenance.

Teaching Methodology:

Lecturing, Written Assignments, Project, Report Writing

Course Assessment:

Midterm exam, Final Exam, Assignments

Reference Materials:

1. Web Engineering, Rajiv Chopra, Prentice-Hall of India, 2016
2. Web Development and Design Foundations with HTML5, Terry Felke-Morris,9th Edition, Pearson ,2018
9. 3. Web-Based Learning: Design, Implementation and Evaluation 2nd edition, Springer,
10. 2018
4. Web Engineering: A Practitioners' Approach, Roger S. Pressman, McGraw Hill, 2008.
11. 5. Learning PHP, MySQL & JavaScript: With jQuery, CSS & HTML5, Robin Nixon , O'Reilly Media; 5 edition, 2018
6. JavaScript: The Definitive Guide, 8th Edition, David Flanagan. O'Reilly Media. 2014.

Software Design and Architecture			
Credit Hours:	3 (3,0)	**Prerequisites:**	Software Engineering

Course Learning Outcomes (CLOs):

1. Understand the role of design and its major activities within the OO software development process, with focus on the Unified process.
2. Comprehend the advantages of consistent and reliable software design.
3. Design OOD models and refine them to reflect implementation details
4. Apply and use UML to visualize and document the design of software systems.
5. Implement the design model using an object-oriented programming language.

Course Content:

Software Design Concepts, Design principles, Object-Oriented Design with UML, System design and software architecture, Object design, Mapping design to code, User interface design, Persistent layer design, Web applications design, State machine diagrams and modeling, Agile software engineering, Design Patterns, Exploring inheritance, Interactive systems with MVC architecture, Software reuse. Architectural design issues, , Software Architecture, Architectural Structures & Styles-, Architectural Patterns, Architectural & Design Qualities, Quality Tactics, Architecture documentation, Architectural Evaluation, Model driven development.

Teaching Methodology:

Lecturing, Written Assignments, Project, Report Writing

Course Assessment:

Midterm exam, Final Exam, Assignments

Reference Materials:

1. Software Engineering: A Practitioner's Approach, Roger S. Pressman, Bruce R. Maxim, 8th Ed, McGraw-Hill Education, 2015.
2. Object-Oriented Analysis, Design and Implementation, Brahma Dathan, Sarnath Ramnath, 2nd Ed, Universities Press, India, 2014.
3. Software Modeling and Design: UML, Use Cases, Patterns, and Software Architectures, Hassan Gomaa, Cambridge University Press, 2011.
4. Head First Design Patterns, Eric Freeman, Elisabeth Freeman, Kathy Sierra and Bert Bates, O'Reilly Media, Inc. 2004.

Software Project Management			
Credit Hours:	3 (3,0)	**Prerequisites:**	Software Engineering

Course Learning Outcomes (CLOs):

1. Explain principles of the project lifecycle and how to identify opportunities to work with learners on relevant and appropriate project scenarios to share this understanding
2. Critically evaluate and discuss the issues around project management and its application in the real world with course participants and learners
3. Choose project management techniques for IT projects to initiate, plan, execute and evaluate a project and work in teams to create a project plan for a project scenario that includes key tasks, critical path, dependencies and a realistic timeline.
4. Present strategies for gaining confidence in managing projects through simple project planning examples.

Course Content:

Introduction to Software Project Management, Project Management concepts, Project Management Tools, PMI‘s Knowledge areas, PMI Framework, PMI Process Groups. Understanding Organizations. Project Planning, Project Evaluation, Selection of an Appropriate Approach in Project, Software Effort Estimation, Activity Planning, Risk Management, Evaluating the Risks to the Schedule, Risk Control, Configuration Management and Maintenance, Environment for Configuration Control, Resource Allocation, Monitoring & Control, Review and Evaluation, Challenges of Outsourcing in Project Management

Teaching Methodology:

Lecturing, Written Assignments, Project, Report Writing

Course Assessment:

Midterm exam, Final Exam, Assignments

Reference Materials:

1. Software Project Management, Bob Hughes and Mike Cotterell, McGraw-Hill Education; 5th Edition (2009).
2. A Guide to the Project Management Body of Knowledge, 6th Edition (PMBOK Guides),2017
3. Mastering Software Project Management: Best Practices, Tools and Techniques, Murali K. Chemuturi and Thomas M. Cagley Jr., J. Ross Publishing, 2010
4. Effective Project Management: Traditional, Agile, Extreme, Robert K. Wysocki, Wiley; 6th Edition, 2011

Software Quality Engineering			
Credit Hours:	3 (3,0)	**Prerequisites:**	Software Engineering

Course Learning Outcomes (CLOs):

1. Outline software testing and software quality assurance principles.
2. Prepare test case and test suites for completely testing all aspects of a system under test (SUT)
3. Analyze which of the software testing techniques are relevant for a particular case and know software reliability analysis tools and techniques.
4. Compile findings of a quality assurance cycle.

Course Content:

Software Quality, Software Quality Attributes, Quality Engineering., Testing: Concepts, Issues, and Techniques, Software testing lifecycle., Testing Scopes., Testing Approaches., Testing Concepts., Test Planning Process, Introduction to testing process, Requirement of software test planning, Testing documentation, Reporting and historical data recording., Software testing techniques, Testing philosophies , Testing strategies, Model based testing, Software testing techniques, Testing using models, Domain and combinatorial testing, Unit and integration testing, Acceptance testing, Test automation, Slicing, Software reliability models and engineering, Introduction, Exponential model., Reliability growth models, Modeling process, Software inspections, Software reviews, Inspection checks and metrics, Quality Models, Models for quality assessment, Product quality metrics, Quality Measurements, In-Process metrics for software testing, In-Process quality management, Effort/outcome models, System testing, Introduction to sub-system testing, From functional to system aspects of testing, System testing, Introduction to system testing, Scenarios development, System testing, Use-cases for testing, Specification-based testing, Open issues on software testing

Teaching Methodology:

Lecturing, Written Assignments, Project, Report Writing

Course Assessment:

Midterm exam, Final Exam, Assignments

Reference Materials:

1. Paul Jorgensen, Software Testing, A Craftsman's Approach, 4th Ed. CRC Press, Taylor and Francis Group, 2015
2. Bernard Homes, Fundamentals of Software Testing, ISTE, Wiley, 2012
3. Software Engineering, “Ian Sommerville, 9th Edition, Addison Wesley, 2011

Software Re-Engineering			
Credit Hours:	3 (3,0)	**Prerequisites:**	Software Construction and Development

Course Learning Outcomes (CLOs):

1. Explain the concepts and technique of software re-engineering.
2. Apply reengineering techniques to maintain and modify software systems
3. Analyze and understand maintenance related problems associated with object oriented software systems.
4. Able to perform complex design reengineering and reverse engineering problems.

Course Content:

Salient topics include the terminology and the processes pertaining to software evolution, fundamental re-engineering techniques to modernize legacy systems including source code analysis, architecture recovery, and code restructuring, software refactoring strategies, migration to Object Oriented platforms, quality issues in re-engineering processes, migration to network-centric environments, and software integration, reverse engineering, program comprehension, source code transformation and refactoring strategies, software maintenance and re-engineering economics.

Teaching Methodology:

Lecturing, Written Assignments, Project, Report Writing

Course Assessment:

Midterm exam, Final Exam, Assignments

Reference Materials:

1. Re-engineering legacy software, David Lorge Parnas, Chris Birchall, Safari Books, Shelter Island, NY, 2016
2. Reengineering, Priyadarshi Tripathy and Kshirasagar Naik, John Wiley & Sons, Inc.2015
3. Software Maintenance and Evolution: a Roadmap, K.H.Bennett and V.T Rajlich, The Future of Software Engineering, ACM Press 2000.

Web Technologies			
Credit Hours:	3 (3,0)	**Prerequisites:**	none

Course Learning Outcomes (CLOs):

Identify the need , use and build and appropriate web platform to solve a given problem in web domain

Course Content:

Introduction to Web Applications, TCP/IP Application Services. Web Servers: Basic Operation, Virtual hosting, Chunked transfers, Caching support, Extensibility. SGML, HTML5, CSS3. XML Languages and Applications: Core XML, XHTML, XHTM MP. Web Service: SOAP, REST, WML, XSL. Web Services: Operations, Processing HTTP Requests, Processing HTTP Responses, Cookie Coordination, Privacy and P3P, Complex HTTP Interactions, Dynamic Content Delivery. Server Configuration. Server Security. Web Browsers Architecture and Processes. Active Browser Pages: JavaScript, DHTML, AJAX. JSON, Approaches to Web Application Development. Programing in any Scripting language. Search Technologies. Search Engine Optimization. XML Query Language, Semantic Web, Future Web Application Framework.

Teaching Methodology:

Lecturing, Written Assignments, Project, Report Writing

Course Assessment:

Midterm exam, Final Exam, Assignments

Reference Materials:

1. Web Application Architecture: Principles, protocols and practices by Leon Shklar and Richard Rosen, Wiley; 2nd Edition (May 5, 2009). ISBN-10:047051860X
2. Web Technologies: A Computer Science Perspective by Jeffrey C. Jackson, Prentice Hall; 1st Edition (August 27, 2006). ISBN-10:0131856030
3. Web Development and Design Foundations with HTML5, Terry Felke-Morris,9th Edition, Pearson ,2018
4. Learning PHP, MySQL & JavaScript: With jQuery, CSS & HTML5, Robin Nixon , O'Reilly Media; 5 edition, 2018

Annexure I

Program Electives (Technical Electives)

Any five courses (15 Credit Hours)

Sr. No.	Course Code	Course Title	Cr. Hrs.
1.	IT4076	Advanced Networks	3
2.	IT4163	Advanced Web Technologies	3
3.	CS4131	Big Data Programming	3
4.	IT4021	Bioinformatics	3
5.	IT4146	Blockchain Technology and Application	3
6.	CS4173	Cloud Computing	3
7.	IT4222	Cognitive Radio Communication and Networks	3
8.	CS4023	Competitive Programming	3
9.	SE4116	Component Based Software Engineering	3
10.	CS4211	Computer Animations	3
11.	CS4012	Computer Graphics	3
12.	CS4013	Computer Vision	3
13.	CS4123	Cyber Forensics	3
14.	IT4073	Data Communications	3
15.	SE4132	Data Mining	3
16.	CS4134	Data Science & Big Data Analytics	3
17.	CS4135	Data Science Technologies	3
18.	IT4143	Data Warehousing	3
19.	IT4144	Database Administration	3
20.	CS4152	Deep Learning and Neural Networks	3
21.	SE4115	Design Pattern and Refactoring	3
22.	CS4014	Digital Image Processing	3
23.	CS4145	Distributed Database Systems	3
24.	IT4174	Distributed Systems	3
25.	CS4033	Embedded Systems	3

26.	SE4164	Enterprise Application Development	3
27.	IT4232	Enterprise Architecture	3
28.	CS4153	Expert Systems and Knowledge Management	3
29.	IT4221	Free Space Optics	3
30.	SE4165	Games Design and Development	3
31.	CS4133	Information Retrieval	3
32.	IT4015	Information Systems	3
33.	CS4176	Internet of Things	3
34.	SE4166	iPhone Applications Development	3
35.	CS4154	Machine Learning	3
36.	SE4167	Mobile Application Development	3
37.	CS4155	Multi-Agent Systems	3
38.	CS4124	Multimedia Security	3
39.	CS4156	Natural Language Processing	3
40.	IT4074	Network Flows	3
41.	IT4223	Next Generation Networks	3
42.	CS4175	Pervasive Smart Environments	3
43.	SE4125	Secure Software Development	3
44.	IT4233	System Integration and Architecture	3
45.	IT3163	Web Technologies	3

Annexure II

Supporting Electives (Any three, 9 Credit Hours)

- Business Process Engineering
- Operation Research
- Formal Methods in Software Engineering
- Analysis of Algorithm
- Simulation and Modeling
- Stochastic Processes

Annexure III

University Electives (non-exhaustive list)

Any four (12 Credit Hours)

- Introduction to Psychology
- Foreign Language
- Business Ethics
- Principles of Marketing
- Introduction to Sociology
- Life & Learning
- Economics